

1 **Title: Deep learning for time series classification in ecology**

2

3 **Authors:** César Capinha<sup>1</sup>, Ana Ceia-Hasse<sup>2</sup>, Andrew M. Kramer<sup>3</sup>, Christiaan Meijer<sup>4</sup>

4 <sup>1</sup>Centro de Estudos Geográficos, Instituto de Geografia e Ordenamento do Território - IGOT,

5 Universidade de Lisboa, Rua Branca Edmée Marques, 1600-276 Lisboa, Portugal.

6 <sup>2</sup>Global Health and Tropical Medicine, Institute of Hygiene and Tropical Medicine, NOVA

7 University of Lisbon, Rua da Junqueira 100, 1349-008 Lisbon, Portugal.

8 <sup>3</sup>Department of Integrative Biology, University of South Florida, Tampa, Florida, USA.

9 <sup>4</sup>Netherlands eScience Center, Science Park 140, 1098 XG Amsterdam, the Netherlands

10

11 **Correspondence author:** César Capinha | E-mail: cesarcapinha@campus.ul.pt

12 **Running headline:** Classify time series with deep learning

13

14

15

16

17

18

19

20

21

22 **Abstract**

23 1. Time series classification consists of assigning time series into one of two or more  
24 predefined classes. This procedure plays a role in a vast number of ecological classification  
25 tasks, including species identification, animal behaviour analysis, predictive mapping, or the  
26 detection of critical transitions in ecological systems. In ecology, the usual approach to time  
27 series classification consists of transforming the time series into static predictors and then  
28 using these in conventional statistical or machine learning models. However, recent deep  
29 learning approaches now enable the classification using the raw time series data, avoiding the  
30 need for domain expertise, eliminating subjective and resource-consuming data  
31 transformation procedures, and potentially improving classification results.

32 2. We here introduce ecologists to time series classification using deep learning models. We  
33 describe some of the deep learning architectures relevant for time series classification and  
34 show how these architectures and their hyper-parameters can be tested and used for the  
35 classification problem at hand. We illustrate the approach using three case studies from  
36 distinct ecological subdisciplines: *i*) species identification from wingbeat spectrograms; *ii*)  
37 species distribution modelling from time series of climatic variables and *iii*) the classification  
38 of phenological phases from continuous meteorological data.

39 3. The deep learning approach delivered ecologically robust and high performing  
40 classifications for the three case studies. The results obtained also allowed us to point future  
41 research directions and highlight current limitations.

42 4. We demonstrate the high potential and wide applicability of deep learning for time series  
43 classification in ecology. We recommend this approach be considered as an alternative to  
44 commonly used techniques requiring the transformation of time series data.

45

46 **Keywords:** AutoML; Classification; Data-driven; Deep learning; Scalability; Sequential  
47 data; Time series

48

#### 49 **Introduction**

50 The recent increase in affordability, capacity, and autonomy of sensor-based technologies  
51 (Peters et al., 2014; Bush et al., 2017), as well as an increasing number of contributions from  
52 citizen scientists and the establishment of international research networks (Hurlbert & Liang,  
53 2012; Bush et al., 2017) is allowing an unprecedented access to time series of interest for  
54 ecological research (Reichstein et al., 2019). A common aim of ecologists using these data  
55 concerns assigning them into predefined classes, such as ecological states or biological  
56 entities. Typical examples include the recognition of bird species from sound recordings (e.g.,  
57 Priyadarshani, Marsland, Juodakis, Castro, & Listanti 2020), the distinction between phases  
58 in the annual life cycle of plants (i.e., ‘phenophases’) from spectral time series (Melaas,  
59 Friedl, & Zhu 2013), or the recognition of behavioural states from animal movement data  
60 (Shamoun-Baranes, Bouten, van Loon, Meijer, & Camphuysen 2016). Many other examples  
61 exist, with scopes of application that range from the molecular level (Jaakkola, Diekhans, &  
62 Haussler 2000) to the global scale (e.g., Schneider, Friedl, & Potere 2010).

63

64 The assignment of time series into one of two or more predefined classes (hereafter referred  
65 to as ‘time series classification’; Keogh and Kasetty 2003) can be performed using a variety  
66 of different approaches, ranging from manual, expert-based, classification (Priyadarshani et  
67 al., 2020) to fully automated procedures (see Bagnall, Lines, Bostrom, Large, & Keogh 2017  
68 for examples). In ecology, time series classification is generally approached by processing the  
69 time series data into a new set of ‘static’ variables – using hand-designed transformations, or  
70 techniques such as Fourier or wavelet transforms – and then using these variables as  
71 predictors in ‘classical’ classification algorithms, such as logistic or multinomial regressions  
72 or random forests (e.g., Reside, VanDerWal, Kutt, & Perkins 2010; Shamoun-Baranes et al.,  
73 2016; Dyderski, Paż, Frelich, & Jagodziński 2017; Capinha, 2019; Priyadarshani et al.,  
74 2020). In machine learning terminology, this approach is known as ‘feature-based’, where the  
75 ‘features’ are the variables that are extracted from the time series.

76

77 Despite the wide adoption of feature-based approaches, important limitations still undermine  
78 their predictive performance and scalability. A key constraint concerns the need for domain-  
79 specific knowledge about the phenomenon that is being classified in order to obtain ‘optimal’  
80 sets of features. While this may not seem limiting, considering the ever-growing body of  
81 knowledge in the ecological literature, in reality few, if any, ecological phenomena are fully  
82 understood (Currie, 2019). This inherently limits and casts doubt about the optimality of  
83 human-mediated selections of ‘relevant’ predictors of their behaviour. This limitation can be  
84 illustrated for species distribution modelling, a popular field among ecological modellers.  
85 These models often rely on readily available sets of predictors that summarize long-term  
86 climate averages and variability, (e.g., the BIOCLIM variables; Booth, Nix, Busby, &

87 Hutchinson 2014), despite recognition that species distributions can also respond to short-  
88 term meteorological variation (e.g., Reside et al., 2010). Accordingly, these common  
89 predictors cannot guarantee a comprehensive representation of the role of climate in  
90 determining the distribution of species. Additionally, scaling modelling frameworks can  
91 result in reliance on pre-processed predictors because performing species-specific feature  
92 extraction could be prohibitively costly, in terms of human and time resources, when  
93 modelling the distribution of hundreds of species.

94

95 Here we discuss and demonstrate the use of supervised deep learning models for time series  
96 classification. Deep learning models are a set of recent, complex architectures of artificial  
97 neural networks (LeCun, Bengio, & Hinton 2015; Christin et al., 2019), which have enabled  
98 significant advances of performance in highly complex tasks, particularly image recognition  
99 (LeCun et al., 2015) – including in ecology (e.g., Christin, Hervet, & Lecomte 2019; Ferreira  
100 et al., in press). Recently, the usefulness of these models for time series classification has  
101 been highlighted (Wang, Yan, & Oates 2017; Fawaz, Forestier, Weber, Idoumghar, & Muller  
102 2019). However, their adoption for this purpose in ecology remains limited (see Sethi et al.  
103 2020, for an exception). A difference between deep learning models and feature-based  
104 approaches is that deep learning models work directly with the raw time series. The  
105 identification of relevant features in the time series is performed by the model itself and is  
106 guided by the contribution that the features have in distinguishing the classes. Accordingly, a  
107 promise of these models is that they may capture relevant information that would be missed if  
108 relying on subjective sets of static features, improving predictive performances. Additionally,

109 because there is no need of human intervention in feature extraction, deep learning models  
110 allow a full, end-to-end, automation of computational workflows.

111

112 We explain deep neural networks and describe some of the modelling architectures more  
113 relevant in the context of classifying time series. Next, we demonstrate the application of  
114 deep learning models for time series classification using three case studies. First, we perform  
115 species identification based on recordings of insect wing flap movements, second, we predict  
116 the potential distribution of a vulnerable mammal species using time series of monthly  
117 climate data, and third we predict the seasonal patterns of fruiting of a mushroom species,  
118 based on meteorological time series. We implement all models using ‘mcfly’ (van Kuppevelt  
119 et al., 2020), a Python package aimed at time series classification for non-experts in deep  
120 learning, and which should be accessible to the generality of ecological modelers.

121

## 122 **Materials and Methods**

### 123 *Deep neural networks for time series classification*

124 Artificial neural networks (ANN) are algorithms inspired by how biological nervous systems  
125 process information. These models are often conceptualised in terms of nodes (or ‘neurons’)  
126 and weighted links. A basic ANN architecture includes a first layer of nodes, representing the  
127 input data, a second (‘hidden’) layer with nodes performing data aggregation followed by  
128 nonlinear transformation, and a final (‘output’) layer where the predicted values are  
129 computed. The nodes in each layer are connected to the nodes in the next layer through  
130 weighted links. Function fitting in ANNs proceeds by iteratively adjusting the weights of  
131 links between the layers. An important notion is the ‘epoch’, which refers to when the entire

132 training dataset is passed forward and backward across the network one time. During each  
133 epoch, the weights are updated to improve the network's predictions, given the information  
134 fed to the input layer. For more details on ANNs see, among others, LeCun et al. (2015) and  
135 references therein.

136

137 'Deep' neural networks refer broadly to ANN architectures that are capable of training large  
138 numbers of hidden layers and neurons (LeCun et al., 2015). This capacity determines the  
139 level of abstraction that the models can attain in representing the input data. Models with  
140 more hidden layers can capture more complex patterns and achieve a deeper hierarchy of  
141 features. In other words, shallow models tend to capture 'basic' patterns (e.g., a 'spike' in a  
142 specific time step), while deeper models are able to 'learn' more complex abstractions (e.g.,  
143 spikes combined with a reduced long-term variability).

144

145 Unlike commonly believed, deep learning models do not always require large amounts of  
146 data for training. For instance, some of these models can provide competitive classification  
147 results with as low as 50 samples (Fawaz et al., 2019).

148

149 Many deep learning architectures can be used for time series classification (Wang et al.,  
150 2017; Fawaz et al., 2019). These architectures differ in the number of layers, and the  
151 mathematical functions the layers perform, as well as in the way information flows between  
152 them. Below we provide a description of four architectures used for time series classification:  
153 Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Residual  
154 Networks (ResNet) and Inception Time Networks (InceptionTime). These architectures were

155 chosen because they are widely adopted for time series classification and because they are  
156 available in mcfly (the software we use here for model implementation; van Kuppevelt et al.,  
157 2020).

158

### 159 *Convolutional Neural Networks*

160 Convolutional neural networks (CNN) are an influential class of deep neural networks. These  
161 networks have been mainly applied for pattern recognition in image data (e.g., Christin et al.,  
162 2019; Ferreira et al., in press), but effective examples of their application for time series  
163 classification have been recently published (e.g., Zhao Lu, Chen, Liu, & Wu 2017). A key  
164 component of CNNs are the so-called convolutional layers (LeCun et al., 2015). These layers  
165 extract local features from the raw time series by applying ‘filters’. Each filter determines if a  
166 given pattern (e.g., ‘a spike’) occurs in the data and in what regions. These layers are often  
167 followed by rectified linear unit (ReLU) (or a similarly shaped function) and ‘pooling’ layers.  
168 The ReLU layers transform the summed weighted input from nodes in the convolutional  
169 layer into outputs that range from 0 to  $+\infty$ , while pooling layers reduce the dimensionality of  
170 outputs from the ReLU layer. CNNs often layer multiple instances of convolution, ReLU and  
171 pooling layers in a sequence, to build a hierarchy of increasingly abstract features. This  
172 sequence of layers is usually followed by a fully connected (or ‘dense’) layer, where each  
173 node is connected to all nodes in adjacent layers, and where classification outputs are  
174 calculated.

175

### 176 *Recurrent Neural Networks*



177 Recurrent neural networks (RNNs) are specifically designed for sequence-type input data,  
178 such as time series (LeCun et al., 2015; Fawaz et al., 2019). These models are defined by  
179 inclusion of feedback loops, where the output of a layer is added to the next input and fed  
180 back into the same layer. This allows RNNs to characterize sequential patterns in the input  
181 data, but their ability to capture long term dependencies is limited due to the RNN's tendency  
182 to prioritize signals in the short term while failing to learn long term signals (i.e., the  
183 'vanishing gradient problem'; Bengio, Simard, & Frasconi 1994). To overcome this problem  
184 several adaptations to the simple RNN architecture have been proposed, the most popular of  
185 which being the use of gating units, such as 'Long Short Term Memory' (LSTM) and 'Gated  
186 Recurrent Units' (GRU) (Chung, Gulcehre, Cho, & Bengio 2014). Gating is a technique that  
187 helps the networks decide to either forget the current input or to remember it for future time  
188 steps, hence effectively improving the modelling of long-term dependencies (Chung et al.,  
189 2014).

190

### 191 *Residual Networks*

192 Residual networks (ResNet) are recently proposed in the context of image recognition (He,  
193 Zhang, Ren, & Sun 2016). Basically, these networks introduce a new type of component, the  
194 'Residual Block', to CNN-type models. The aim of these blocks is to allow the training of  
195 deeper models (i.e., having more hidden layers). In theory, deeper models should improve  
196 classification performances, as they allow higher levels of data abstraction. However, in  
197 practice the performances may not improve, among other things, due to the vanishing  
198 gradient problem (see above). The use of residual blocks aims to address this by forwarding  
199 the output of layers directly into layers that are several levels deeper (e.g., 2–3 layers ahead).

200 Recently, this architecture has been applied for time series classification (Wang et al., 2017),  
201 often performing very well (Fawaz et al., 2019).

202

### 203 *Inception Time Networks*

204 Inception time networks are a very recent type of architecture, proposed specifically for time  
205 series classification (Fawaz et al., 2019). This network is an ensemble of CNN models having  
206 ResNet-type components and modules called ‘inceptions’. Inception modules ‘rework’ how  
207 convolution layers act in the networks, so that instead of being stacked sequentially, they are  
208 ordered to work on the same level in parallel. This approach allows the application of  
209 multiple filters with highly varying temporal lengths working on the same input time series.  
210 In comparison to sequential convolutional layers (as in ‘simple’ CNN) this lowers processing  
211 costs and reduces the risk of fitting noise in the data (i.e., overfitting) (Fawaz et al., 2019).

212

### 213 *The mcfly Python library*

214 Deep learning models can be implemented using several programming languages and  
215 specialised libraries (see Christin et al., 2019 for a review). Here, we use mcfly, a Python  
216 package for time series classification using deep learning (van Kuppevelt et al., 2020). This  
217 package is aimed at non-experts and it should be easy to use for ‘mid-level’ ecological  
218 modellers. Mcfly also delivers a standardized workflow that ‘generates’ distinct, ready-to-  
219 train models and tests which is best suited for the classification task. This assists non-experts  
220 in deep learning in identifying a suitable modelling architecture and implementing the model  
221 from scratch (Christin et al., 2019).

222

223 Mcfly utilizes TensorFlow ([www.tensorflow.org](http://www.tensorflow.org)) an extensively adopted machine learning  
224 library, it can make use of (but does not require) dedicated hardware (such as Graphical  
225 Processing Units: ‘GPUs’), works with both univariate and multivariate time series (‘single  
226 channel’ and ‘multichannel data’, in machine learning terminology) and includes procedures  
227 for inspecting and visualizing the parameters of trained models. In its current version (v.3.0)  
228 mcfly generates CNN, Deep convolutional LSTM (‘DeepConvLSTM’; an architecture  
229 composed of convolutional and LSTM recurrent layers), ResNet and InceptionTime  
230 architectures. Specific details about the components and structure of each architecture are  
231 given in van Kuppevelt et al. (2020).

232

233 Model selection in mcfly proceeds by generating a set of candidate models with architectures  
234 and hyperparameters (e.g., number of layers; learning rate) selected at random from a  
235 prespecified range of values (see Figure 1). Each candidate model is trained using a small  
236 subset of the data (data partition  $A_t$ ; Figure 1) during a small number of epochs. After  
237 training, the performance of the candidate models is compared using a left-out validation data  
238 set ( $A_v$ ; Figure 1). The selected candidate model (usually the best performing among  
239 candidates) is then trained on the full training data ( $B_t$ ; Figure 1). In this step it is required to  
240 identify an optimal number of training epochs, to avoid under- or overfitting of the model. A  
241 model trained too few epochs will not capture all relevant patterns in the data, reducing  
242 predictive performance. A model trained for an excessive number of epochs might overfit,  
243 reducing its generality and ability to classify new data. There is no definitive way to identify  
244 an optimal number of training epochs, but one practical approach is through monitoring the  
245 model’s validation performance (i.e., using holdout data partition  $B_v$ ; Figure 1). The

246 ‘optimal’ number of training epochs is the one that provides the best validation performance.

247 Finally, the performance of the model having an ‘optimal’ number of training epochs is

248 evaluated using a ‘final’ test data set ( $T$ ; Figure 1), providing the best estimate of the

249 predictive performance of the model.

250

251 For the three case studies below, we used the same model generation and selection strategy.

252 We had *mcfly* generate 20 candidate models, five for each architecture type. These models

253 were trained during 4 epochs (using  $A_t$ ). The candidate model achieving highest performance

254 in predicting the classes of the validation data ( $A_v$ ) was then trained on the full training data

255 set ( $B_t$ ). For each epoch we measured training performance, as provided by *mcfly* (which

256 uses the accuracy metric i.e., ‘the proportion of cases correctly classified’). The classification

257 performance on the validation data ( $B_v$ ) was measured using the area under the receiver

258 operating characteristic curve (AUC), a metric that is not affected by differences in the

259 prevalence of classes and is widely used in ecology (e.g., Dyderski et al., 2017).

260

261 To identify an ‘optimal’ number of training epochs, we examined the progression of

262 validation performance ( $B_v$ ). Models can be trained for an infinite number of epochs, so here

263 we stopped training if no increase in validation performance was observed after 25 epochs

264 (other thresholds could be considered, according to time resources available). Finally, the

265 model trained with the number of epochs showing highest AUC in predicting  $B_v$  was used to

266 classify the test data (data set  $T$ ), with performance measured using AUC.

267

268 We recorded processing time of all models from the onset of training of candidate models to  
269 the last training epoch evaluated for the selected model. This was done on two distinct  
270 systems: a ‘desktop PC’ with an Intel i7 4-Core (3.40GHz) processor and 8GB RAM and a  
271 ‘high-end workstation’ with an AMD Ryzen 9 12-Core (3.80 GHz) processor, 64 GB RAM  
272 and an NVidia RTX 2060 GPU. Because CPU- and GPU-based TensorFlow generate distinct  
273 random hyperparameters, modelling results will differ between the two computer systems.  
274 We report results and processing times for the desktop PC system. For the workstation we  
275 report processing time only. We emphasize that the timings recorded in the two systems are  
276 not directly comparable as they correspond to distinct modelling routes.

277

278 It is important to bear in mind that the modelling strategy described aims at general  
279 applicability and further tailoring for specific classification tasks could be beneficial. For  
280 instance, with *a priori* knowledge that a specific architecture, say CNN, is best suited for the  
281 classification task at hand (see discussion section), the selection could be adjusted to generate  
282 only CNN-type candidate models. Further information about fine-tuning of mcfly model  
283 generation and selection can be found in van Kuppevelt et al. (2020).

284

#### 285 *Case study 1: Species identification*

286 In this case study we predict the identity of three insect species: the olive fruit fly (*Bactrocera*  
287 *oleae*), the western honey bee (*Apis mellifera*), and the black fig fly (*Lonchaea aristella*)  
288 using wingbeat spectrograms (frequency series of amplitude values; Potamitis, Rigakis, &  
289 Fysarakis 2015). *B. oleae* is an olive fruit fly pest, which if left unmanaged can lead to large  
290 economic costs worldwide (Potamitis et al., 2015). The wingbeat spectrum characteristics of

291 these three species allow us to exemplify an ‘easy’ classification case and a ‘difficult’  
292 classification case: while in *A. mellifera* harmonics partially overlap with those of *B. oleae*,  
293 these species show differences in frequencies - including the fundamental frequency - and  
294 thus constitute the ‘easy’ classification case; in contrast, *L. aristella* has a wingbeat spectrum  
295 that completely overlaps with that of *B. oleae*, representing the ‘difficult’ classification case.

296

297 We thus have three classes, each corresponding to a species ‘positive’ identity. The data are  
298 balanced (i.e. the number of samples per class is similar) and consist of 230 samples for *B.*  
299 *oleae*, 205 for *A. mellifera*, and 252 for *L. aristella*.

300

301 Species were identified (classified) according to their wingbeat spectrograms, which consist  
302 of frequency series of amplitudes (the predictor variable) obtained from Potamitis et al.  
303 (2015). A sample was composed of a total of 256 steps (frequencies), each step  
304 corresponding to an amplitude value for a frequency. This case study illustrates the use of  
305 these models using only one predictor variable (i.e., a single time series).

306

307 The records of species identity data and predictor variable (amplitude per frequency) were  
308 split into: data for training candidate models (~50%; *A<sub>t</sub>*), data for validating candidate models  
309 (~20%; *A<sub>v</sub>*), data for training the selected model (~70%; *B<sub>t</sub>*; resulting from merging the two  
310 previous data sets), validation data for determining the number of epochs for training the  
311 selected model (~15%; *B<sub>v</sub>*) and test data for final assessment of classification performance  
312 (~15%; *T* in Fig. 1).

313

314 *Case study 2: Species distribution model*

315 In this case study we predict the potential distribution of the Iberian Desman (*Galemys*  
316 *pyrenaicus*) using time series of environmental data. The Iberian Desman is a vulnerable  
317 semi-aquatic species, endemic to the Iberian Peninsula and the Pyrenean Mountains. We  
318 collected distribution records from the Portuguese and Spanish atlases of mammals (Palomo,  
319 Gisbert, & Blanco 2007; Bencatel, Álvares, Moura, & Barbosa 2017). The data consists of  
320 6141 UTM grid cells of 10×10 km, of which 659 record the species presence (class  
321 ‘Presence’) and 5482 its absence (class ‘Absence’).

322

323 The environmental conditions in each cell were characterized using four variables: 1)  
324 maximum temperature; 2) minimum temperature, 3) accumulated precipitation, and 4)  
325 altitude. The first three variables consist of time series of monthly values collected from  
326 CHELSA (Karger et al., 2017) spanning 1989 to 2013, totalling 300 time steps. The fourth  
327 variable was from Fick and Hijmans (2017) and corresponds to temporally invariant values of  
328 altitude (demonstrating inclusion of temporally static predictors), coded as a time series.

329

330 Species distribution data and predictors were split similarly as above with different  
331 proportions: a)  $A_t$  ~ 35%, b)  $A_v$  ~ 35%, c)  $B_t$  ~70%; resulting from merging  $A_t$  and  $A_v$ , d)  
332  $B_v$  ~ 15%, and e) test data set T ~15%. The low percentage of data used for training the  
333 candidate models in comparison to case study 1 aims to reduce computer processing time,  
334 given larger data volume.

335

336 The training and internal validation of deep learning models are sensitive to class imbalance  
337 (i.e., when one or several classes have a much higher number of samples). Strong class  
338 imbalance can bias models towards the prediction of majority classes (Menardi & Torelli,  
339 2014) and reduces the reliability of performance metrics such as accuracy *sensu stricto* (i.e.,  
340 the proportion of correct predictions to the total number of samples), which is used for the  
341 automated selection of candidate models in mcfly (van Kuppevelt et al., 2020). Accordingly,  
342 we balanced our data by randomly duplicating presence records and deleting absence records  
343 until a balance of ~50:50 is obtained, which was executed using the ROSE package  
344 (Lunardon, Menardi, & Torelli 2014) for R (R Core Team, 2020). This was done for the data  
345 sets that mcfly uses for internal assessment of accuracy *s.s.* ( $A_t$ ,  $A_v$  and  $B_t$ , Figure 1). Data  
346 partitioning was performed prior to balancing, to avoid inclusion of replicated cases of the  
347 same data across multiple partitions. The remaining data sets (i.e.,  $B_v$  and  $T$ ) were not  
348 balanced.

349

### 350 *Case study 3: Phenological prediction*

351 In this case study we predict the timing of fruiting of the Parasol mushroom (*Macrolepiota*  
352 *procera*) across Europe. This species produces fruiting bodies valued for human consumption  
353 (Capinha 2019) and predicting their emergence could be useful for managing human pressure  
354 on the species and its habitats. Data is from Capinha (2019), a study employing a feature-  
355 based approach to achieve an equivalent aim. The data have two classes. One class  
356 ('fruiting') corresponds to locations and dates of observation of fruiting bodies of the species  
357 (from 2009 to 2015). The second class corresponds to 'temporal pseudo-absences', which are  
358 records in the same locations of the observation records, but with dates selected at random



359 along the temporal range of the study (Capinha 2019). The aim of the classification is to  
360 distinguish the meteorological conditions associated with the observation of fruiting bodies of  
361 the species from the range of meteorological conditions that are available to it.

362

363 We characterized each record using four time series: 1) mean daily temperature for the  
364 preceding 365 days, 2) daily total precipitation for the preceding 365 days, 3) latitude and 4)  
365 longitude. Time series of temperature and precipitation were extracted from the daily  
366 AGRI4CAST maps (<http://agri4cast.jrc.ec.europa.eu/>), at a cell resolution of 25x25 km.  
367 Geographical coordinates were coded as temporally invariant time series.

368

369 Records from 2009 to 2014 were randomly partitioned into: *At*: 15%, *Av*: 70%, *Bv*: 15%,  
370 and *Bt*: 85% (merging *At* and *Av*). Data for the year 2015 was used to evaluate the predictive  
371 performance of the final model (T), allowing comparison with the performance results  
372 achieved in Capinha (2019).

373

374 To increase the representation of the meteorological conditions occurring in the location of  
375 each observation record, the data consists of 15 pseudo-absence records per each observation  
376 record (Capinha, 2019). Similarly to the previous case study, we corrected for class  
377 imbalance by balancing the number of samples in each class using a random deletion and  
378 duplication approach (Lunardon et al., 2014). This balancing was performed for data sets *At*,  
379 *Av* and *Bt*. Data sets *Bv* and T remained unchanged.

380

381 **Results**

382 *Species identification*

383 The candidate model with greatest ability to distinguish between the spectrograms of the  
384 three insect wingbeats had an InceptionTime architecture (accuracy = 0.85; model number  
385 15; Figure 2b). On the training data set this model showed a progressively increasing training  
386 accuracy with number of epochs (Figure 2c). However, its evaluation against left-out data  
387 (Bv data set; Figure 1) showed that best performances were found mainly between training  
388 epoch ~30 and ~50 ('validation AUC'; Figure 2c), followed by little change. The highest  
389 validation performance was obtained after 47 training epochs. On the test data (T; Figure 1),  
390 this model achieved an average AUC of 0.96, resulting from an AUC of 1 in classifying  
391 between *B. oleae* and *A. mellifera*, an AUC of 0.88 in classifying between *B. oleae* and *L.*  
392 *aristella* and an AUC of 1 in classifying between *A. mellifera* and *L. aristella*. Computer  
393 processing time, from the onset of candidate model training to the 72<sup>nd</sup> training epoch of the  
394 selected model, took 26 minutes on a desktop PC. On the high-end workstation, a distinct  
395 modelling event took 3 minutes.

396

397 *Species distribution model*

398 The best performing candidate model for this case study had a CNN-type architecture (model  
399 number 4; Figure 3b), reaching 0.82 of validation accuracy. On the full training data set, the  
400 model showed a slowly increasing trend of training accuracy with number of epochs (Figure  
401 3c). However, left-out validation data (Bv) showed a decreasing trend of performance after  
402 the ~60<sup>th</sup> epoch ('validation AUC'; Figure 3c), with highest performing classification at the  
403 56<sup>th</sup> training epoch. The model trained with this number of epochs achieved an AUC of 0.95  
404 on the final test data (T). Most of northern Iberian Peninsula was predicted as suitable to the

405 Iberian Desman, particularly the high mountainous areas (Figure 3e). Computer processing  
406 time took 2 hours and 49 minutes on a desktop PC. A distinct modelling event on the high-  
407 end workstation took 19 minutes.

408

#### 409 *Phenological prediction*

410 For this case study, the selected candidate model had an InceptionTime-type of architecture  
411 (model number 2; Figure 4a), achieving 0.81 validation accuracy. This model rapidly  
412 increased in training accuracy, but its classification performance measured with external data  
413 increased only up to the 5<sup>th</sup> epoch (Figure 4b). The model trained for 5 epochs achieved an  
414 AUC of 0.91 on the final test data. The predicted probabilities of fruiting for an example site  
415 (Figure 4c) show the ability of the model to capturing seasonal variation, with higher  
416 probabilities generally being predicted for the Autumn season, but with important inter-  
417 annual differences. Computer processing time took 10 hours and 23 minutes on a desktop PC.  
418 On a high-end workstation a distinct modelling event took 18 minutes.

419

#### 420 **Discussion**

421 Deep artificial neural networks are a flexible modelling technique with notable success in a  
422 range of scientific fields (LeCun et al., 2015). In ecology, the adoption of these models is still  
423 in its infancy and has been mainly directed towards image recognition (Christin et al., 2019;  
424 Ferreira et al., 2020). We here introduce the use of deep learning models for time series  
425 classification and demonstrate how these models can be implemented and evaluated for  
426 distinct tasks across subfields of ecology.

427

428 Our case studies demonstrate the versatility and potential of deep learning for time series  
429 classification. In the first case study, an InceptionTime model performed well in  
430 distinguishing insect species based on spectrograms of their wingbeats. Given the use of  
431 different data partition strategies and performance metrics, the performance measured for this  
432 model is not fully comparable to those obtained by Potamitis et al. (2015) – who classified  
433 the same data using distance and feature based approaches. However, our study more  
434 accurately identified the honeybee, suggesting its superior classification ability. In the case of  
435 the Iberian desman the predictions from a CNN model also achieved a very high  
436 performance, and the predicted spatial patterns are congruent with the known distribution of  
437 the species and with existing predictions from ‘classic’ feature-based approaches (Barbosa,  
438 Real, & Vargas 2011). Finally, an InceptionTime model projected ecologically plausible  
439 patterns of fruiting seasonality for *Macrolepiota procera*, with performance equaling that  
440 obtained by Capinha (2019) (i.e., an AUC of 0.91 on predictions of fruiting in 2015). Unlike  
441 the raw time series used by deep learning models, Capinha (2019) used a large set ( $n=40$ ) of  
442 hand-crafted features reliant on domain-expertise (e.g., growing degree days).

443

444 Despite the valuable results described above, the advantages of deep learning models for time  
445 series classification in ecology can only be fully appreciated with wider testing, including  
446 different classification tasks and data settings. The benchmarking of classification  
447 performances against traditional modelling approaches and the identification of factors  
448 associated with performance differences (e.g., degree of *a priori* ecological knowledge;  
449 complexity of the phenomena; volume of training data, etc.) will be of paramount  
450 importance. Research efforts should attempt to identify the deep learning architectures and

451 hyperparameters that are best suited for specific ecological phenomena and data types. Thus  
452 far, classification performances from distinct deep learning typologies were compared using  
453 time series data coming from multiple domains (e.g., Fawaz et al., 2019), and the relevance  
454 of these results to ecology remains uncertain.

455

456 A distinctive feature of deep learning approaches is that they allow classifying phenomena  
457 directly from raw time series data. For ecologists, this ability should be seen not merely as a  
458 methodological particularity, but as a conceptual and operational upgrade from traditional  
459 modelling approaches. On one hand, the use of time series data as predictors positively forces  
460 ecologists to consider the temporal component of the analysed phenomena (Wolkovich,  
461 Cook, McLauchlan, & Davies 2014) and, on the other, it relieves them from subjective  
462 decisions about the temporal extent to summarize in static predictors. This reorientation in  
463 thinking was, perhaps, best illustrated by using time series – instead of the usual time-  
464 averaged variables – for predicting the potential distribution of a species. This ‘fully’  
465 temporally explicit approach can be exploited for virtually any ecological or biological entity  
466 or state, as long as the putative drivers have a temporal representation. Further, the usage of  
467 time series data by deep learning models matches the increasing number of high frequency  
468 streams of digital data coming from distinct sources (e.g., satellite sensors, meteorological  
469 stations). The direct integration of these data into the models eliminates the need for resource  
470 consuming feature extraction procedures and is well-suited for operational frameworks aimed  
471 at short-term forecasting (e.g., of algal blooms or disease vector abundances), allowing a  
472 rapid detection of situations of concern.

473

474 As for any modelling approach, deep learning models have limitations. Two obstacles are  
475 particularly prominent: the interpretability of models and computational demand. Limitations  
476 to the interpretation of deep learning models have been well described in the literature (e.g.,  
477 Reichstein et al., 2019), however, they are caused mainly by a lack of available tools. Very  
478 recently important efforts towards the interpretability of deep learning models have been  
479 made (e.g., Siddiqui, Mercier, Munir, Dengel, & Ahmed 2019) and given the fast pace of  
480 deep learning research, we expect that soon deep learning models will be no harder to  
481 interpret than many traditional machine learning models. The challenges arising from  
482 computational demand are harder to solve. Here we showed that ‘typical’ classification tasks  
483 can take several hours to run on a standard desktop computer. Additionally, the  
484 computational expensiveness of deep learning is expected to grow in the future (Thompson,  
485 Greenewald, Lee, & Manso, 2020). To face this challenge, ecologists will likely have to  
486 move in the same direction as their fellow computer scientists and embrace faster hardware  
487 (e.g., GPUs, ‘tensor processing units’ and large-resourced cloud computing services) and  
488 scalable model implementations (e.g., distributed computing).

489

490 In conclusion, we suggest that the use of deep learning for classifying ecological time series  
491 could bring considerable improvements over conventional approaches. Software tools now  
492 exist that allow overcoming the implementation barrier for non-experts and state-of-the-art  
493 classification results seem a reasonable expectation for several tasks. However, only with  
494 extensive testing can the value of this approach be fully recognized. Those willing to venture  
495 through this modelling route could use the data and code we provide as a starting point.

496

497 **Acknowledgments**

498 CC and ACH were supported by Portuguese National Funds through Fundação para a Ciência  
499 e a Tecnologia (CC: CEECIND/02037/2017, UIDB/00295/2020 and UIDP/00295/2020;  
500 ACH: PTDC/SAU-PUB/30089/2017 and GHTM UID/Multi/04413/2013).

501

502 **Author Contributions**

503 CC conceived the ideas and designed methodology; CC and ACH collected and analysed the  
504 data; CC led the writing of the manuscript. All authors contributed critically to the drafts and  
505 gave final approval for publication.

506

507 **Data Availability**

508 Data and code for this study are available from: <https://doi.org/10.5281/zenodo.4017750>

509

510 **References**

- 511 Bagnall, A., Lines, J., Bostrom, A., Large, J., & Keogh, E. (2017). The great time series  
512 classification bake off: a review and experimental evaluation of recent algorithmic  
513 advances. *Data Mining and Knowledge Discovery*, 31(3), 606–660. doi:10.1007/s10618-  
514 016-0483-9
- 515 Barbosa, A. M., Real, R., & Vargas, M. J. (2009). Transferability of environmental  
516 favourability models in geographic space: The case of the Iberian desman (*Galemys*  
517 *pyrenaicus*) in Portugal and Spain. *Ecological Modelling*, 220(5), 747–754.  
518 doi:10.1016/j.ecolmodel.2008.12.004

- 519 Bencatel, J., Álvares, F., Moura, A. E., & Barbosa, A. M. (2017). *Atlas de Mamíferos de*  
520 *Portugal*. Universidade de Évora.
- 521 Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with  
522 gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157–166.  
523 doi:10.1109/72.279181
- 524 Booth, T. H., Nix, H. A., Busby, J. R., & Hutchinson, M. F. (2014). bioclim: the first species  
525 distribution modelling package, its early applications and relevance to most current  
526 MaxEnt studies. *Diversity and Distributions*, 20(1), 1–9. doi:10.1111/ddi.12144
- 527 Bush, A., Sollmann, R., Wilting, A., Bohmann, K., Cole, B., Balzter, H., ... Yu, D. W.  
528 (2017). Connecting Earth observation to high-throughput biodiversity data. *Nature*  
529 *Ecology & Evolution*, 1(7), 1–9. doi:10.1038/s41559-017-0176
- 530 Capinha, C. (2019). Predicting the timing of ecological phenomena using dates of species  
531 occurrence records: a methodological approach and test case with mushrooms.  
532 *International Journal of Biometeorology*, 63(8), 1015–1024. doi:10.1007/s00484-019-  
533 01714-0
- 534 Christin, S., Hervet, É., & Lecomte, N. (2019). Applications for deep learning in ecology.  
535 *Methods in Ecology and Evolution*, 10(10), 1632–1644. doi:10.1111/2041-210X.13256
- 536 Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated  
537 Recurrent Neural Networks on Sequence Modeling. *ArXiv:1412.3555 [Cs]*. Retrieved  
538 from <http://arxiv.org/abs/1412.3555>
- 539 Currie, D. J. (2019). Where Newton might have taken ecology. *Global Ecology and*  
540 *Biogeography*, 28(1), 18–27. doi:10.1111/geb.12842



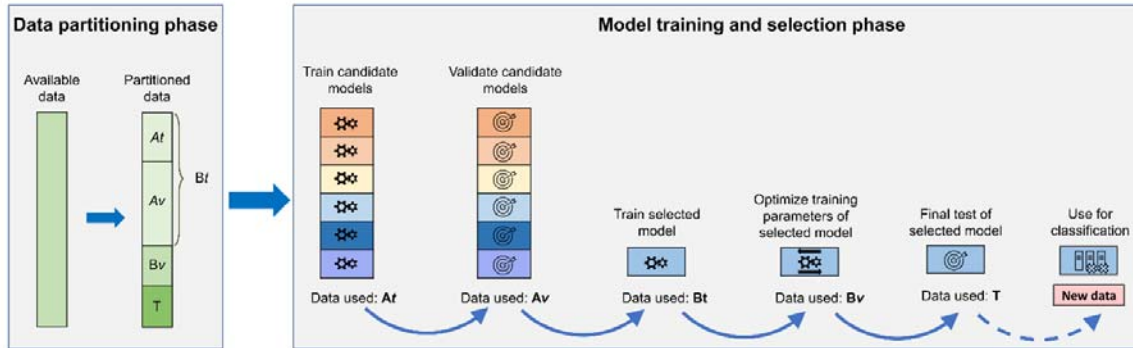
- 541 Dyderski, M. K., Paż, S., Frelich, L. E., & Jagodziński, A. M. (2018). How much does  
542 climate change threaten European forest tree species distributions? *Global Change*  
543 *Biology*, 24(3), 1150–1163. doi:10.1111/gcb.13925
- 544 Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning  
545 for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4),  
546 917–963. doi:10.1007/s10618-019-00619-1
- 547 Ferreira, A. C., Silva, L. R., Renna, F., Brandl, H. B., Renoult, J. P., Farine, D. R., ...  
548 Doutrelant, C. (2020). Deep learning-based methods for individual recognition in small  
549 birds. *Methods in Ecology and Evolution*. 11(9), 1072–1085. doi:10.1111/2041-  
550 210X.13436
- 551 Fick, S. E., & Hijmans, R. J. (2017). WorldClim 2: new 1 km spatial resolution climate  
552 surfaces for global land areas. *International Journal of Climatology*, 37(12), 4302–4315.
- 553 He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition.  
554 In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–  
555 778). doi:10.1109/CVPR.2016.90
- 556 Hurlbert, A. H., & Liang, Z. (2012). Spatiotemporal Variation in Avian Migration  
557 Phenology: Citizen Science Reveals Effects of Climate Change. *PLOS ONE*, 7(2), e31662.  
558 doi:10.1371/journal.pone.0031662
- 559 Jaakkola, T., Diekhans, M., & Haussler, D. (2000). A discriminative framework for detecting  
560 remote protein homologies. *Journal of Computational Biology*, 7(1–2), 95–114.  
561 doi:10.1089/10665270050081405

- 562 Karger, D. N., Conrad, O., Böhrner, J., Kawohl, T., Kreft, H., Soria-Auza, R. W., ... Kessler,  
563 M. (2017). Climatologies at high resolution for the earth's land surface areas. *Scientific*  
564 *Data*, 4, 170122.
- 565 Keogh, E., & Kasetty, S. (2003). On the Need for Time Series Data Mining Benchmarks: A  
566 Survey and Empirical Demonstration. *Data Mining and Knowledge Discovery*, 7(4), 349–  
567 371. doi:10.1023/A:1024988512476
- 568 LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.  
569 doi:10.1038/nature14539
- 570 Lunardon, N., Menardi, G., & Torelli, N. (2014). ROSE: A Package for Binary Imbalanced  
571 Learning. *The R Journal*, 6(1), 79–89.
- 572 Melaas, E. K., Friedl, M. A., & Zhu, Z. (2013). Detecting interannual variation in deciduous  
573 broadleaf forest phenology using Landsat TM/ETM+ data. *Remote Sensing of*  
574 *Environment*, 132, 176–185. doi:10.1016/j.rse.2013.01.011
- 575 Menardi, G., & Torelli, N. (2014). Training and assessing classification rules with  
576 imbalanced data. *Data Mining and Knowledge Discovery*, 28(1), 92–122.
- 577 Palomo, L. J., Gisbert, J., & Blanco, J. C. (2007). *Atlas y Libro Rojo de los Mamíferos*  
578 *Terrestres de España*. Madrid: Organismo Autonomo de Parques Nacionales.
- 579 Peters, D. P. C., Havstad, K. M., Cushing, J., Tweedie, C., Fuentes, O., & Villanueva-  
580 Rosales, N. (2014). Harnessing the power of big data: infusing the scientific method with  
581 machine learning to transform ecology. *Ecosphere*, 5(6), art67. doi:10.1890/ES13-00359.1
- 582 Potamitis, I., Rigakis, I., & Fysarakis, K. (2015). Insect Biometrics: Optoacoustic Signal  
583 Processing and Its Applications to Remote Monitoring of McPhail Type Traps. *PLOS*  
584 *ONE*, 10(11), e0140474. doi:10.1371/journal.pone.0140474

- 585 Priyadarshani, N., Marsland, S., Juodakis, J., Castro, I., & Listanti, V. (2020). Wavelet filters  
586 for automated recognition of birdsong in long-time field recordings. *Methods in Ecology  
587 and Evolution*, *11*(3), 403–417. doi:10.1111/2041-210X.13357
- 588 R Core Team. (2020). A language and environment for statistical computing. Vienna,  
589 Austria: R Foundation for Statistical Computing. Retrieved from  
590 <https://www.r-project.org>
- 591 Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., &  
592 Prabhat. (2019). Deep learning and process understanding for data-driven Earth system  
593 science. *Nature*, *566*(7743), 195–204. doi:10.1038/s41586-019-0912-1
- 594 Reside, A. E., VanDerWal, J. J., Kutt, A. S., & Perkins, G. C. (2010). Weather, Not Climate,  
595 Defines Distributions of Vagile Bird Species. *PLOS ONE*, *5*(10), e13569.  
596 doi:10.1371/journal.pone.0013569
- 597 Schneider, A., Friedl, M. A., & Potere, D. (2010). Mapping global urban areas using MODIS  
598 500-m data: New methods and datasets based on ‘urban ecoregions’. *Remote Sensing of  
599 Environment*, *114*(8), 1733–1746. doi:10.1016/j.rse.2010.03.003
- 600 Sethi, S. S., Jones, N. S., Fulcher, B. D., Picinali, L., Clink, D. J., Klinck, H., ... Ewers, R.  
601 M. (2020). Characterizing soundscapes across diverse ecosystems using a universal  
602 acoustic feature set. *Proceedings of the National Academy of Sciences*, *117*(29), 17049–  
603 17055. doi:10.1073/pnas.2004702117
- 604 Shamoun-Baranes, J., Bouten, W., van Loon, E. E., Meijer, C., & Camphuysen, C. J. (2016).  
605 Flap or soar? How a flight generalist responds to its aerial environment. *Philosophical  
606 Transactions of the Royal Society B: Biological Sciences*, *371*(1704), 20150395.  
607 doi:10.1098/rstb.2015.0395

- 608 Siddiqui, S. A., Mercier, D., Munir, M., Dengel, A., & Ahmed, S. (2019). Tsviz:  
609 Demystification of deep learning models for time-series analysis. *IEEE Access*, 7, 67027–  
610 67040.
- 611 Thompson, N. C., Greenewald, K., Lee, K., & Manso, G. F. (2020). The Computational  
612 Limits of Deep Learning. *ArXiv Preprint ArXiv:2007.05558*.
- 613 van Kuppevelt, D., Meijer, C., Huber, F., van der Ploeg, A., Georgievska, S., & van Hees, V.  
614 T. (2020). Mcfly: Automated deep learning on time series. *SoftwareX*, 12, 100548.  
615 doi:10.1016/j.softx.2020.100548
- 616 Wang, Z., Yan, W., & Oates, T. (2017). Time series classification from scratch with deep  
617 neural networks: A strong baseline. In *2017 International Joint Conference on Neural*  
618 *Networks (IJCNN)* (pp. 1578–1585). doi:10.1109/IJCNN.2017.7966039
- 619 Wolkovich, E. M., Cook, B. I., McLauchlan, K. K., & Davies, T. J. (2014). Temporal  
620 ecology in the Anthropocene. *Ecology Letters*, 17(11), 1365–1379.
- 621 Zhao, B., Lu, H., Chen, S., Liu, J., & Wu, D. (2017). Convolutional neural networks for time  
622 series classification. *Journal of Systems Engineering and Electronics*, 28(1), 162–169.  
623 doi:10.21629/JSEE.2017.01.18

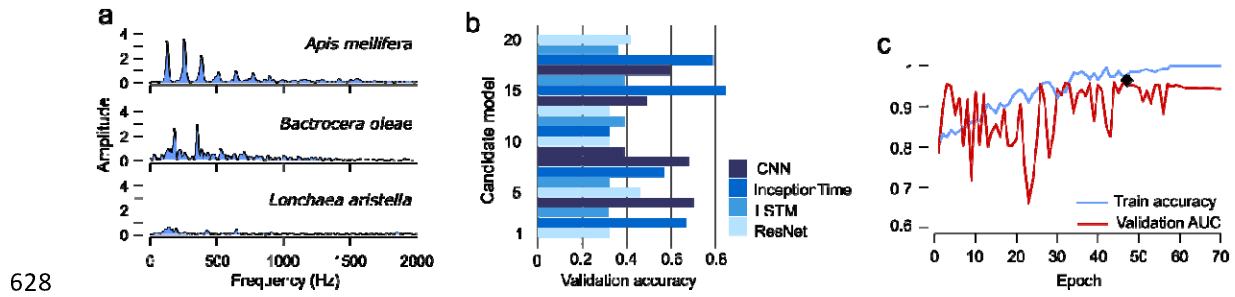
624 **Figures**



625

626 **Figure 1.** Schematic of data partitions and modelling workflow used by the 'mcfly' Python

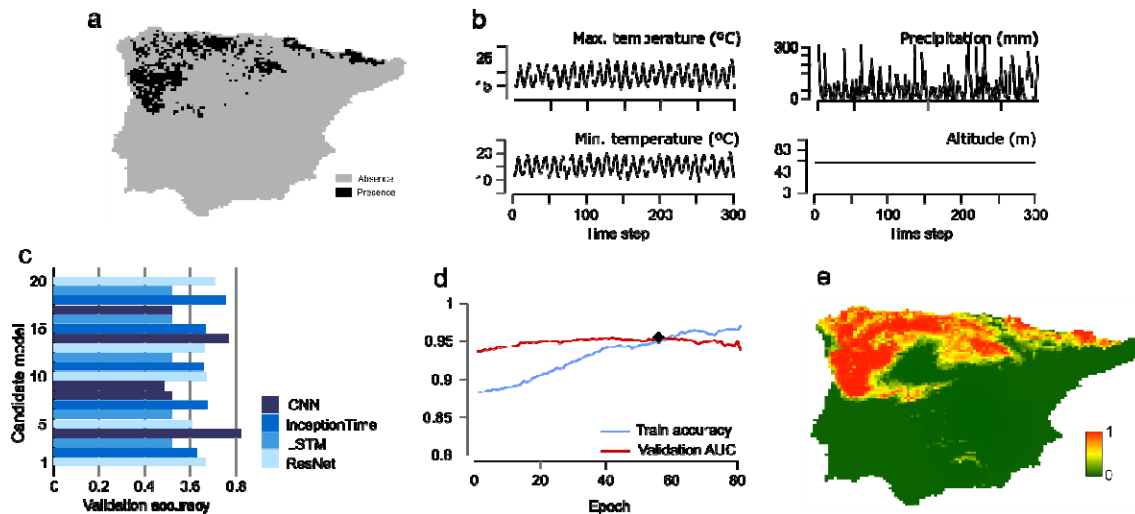
627 package for time series classification.



628

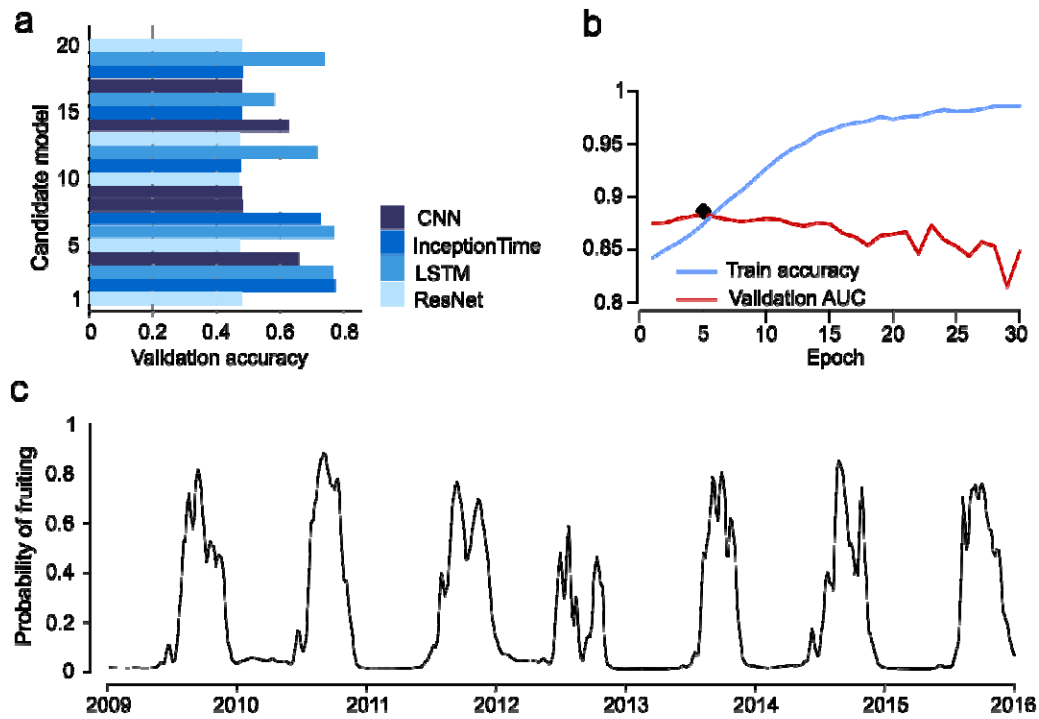
629 **Figure 2.** Data and results of deep learning models classifying insect species from wingbeat  
630 spectrograms. (a) Example wingbeat spectrograms for each species. (b) Validation accuracy  
631 for candidate deep learning models. (c) Training and validation curves of the selected model  
632 along time (highest validation performance is marked with a diamond symbol).

633



634

635 **Figure 3.** Data and results of deep learning models classifying environmental suitability for  
636 the Iberian desman. (a) Presence and absence data of the species. (b) Example of time series  
637 used as predictors. (c) Validation accuracy for candidate deep learning models. (d) Training  
638 and validation curves of the selected model along time. The diamond symbol marks the  
639 highest validation performance. (e) Environmental suitability predicted by the selected  
640 model.



641

642 **Figure 4.** Data and results of deep learning models classifying the fruiting phenology of the  
643 parasol mushroom based on meteorological variation. (a) Validation accuracy for candidate  
644 deep learning models. (b) Training and validation curves of the selected model along time  
645 (the diamond symbol marks the highest validation performance). (c) Patterns of fruiting  
646 seasonality predicted by the selected model for an example location.